

E. Funcionamiento de los Motores

Existen varias formas en las que podemos querer mover el mBot2. Las velocidades de los motores hacia adelante están entre **0 y 100**. Las velocidades hacia atrás están entre **0 y -100**. El movimiento sigue ocurriendo incluso a velocidades muy cercanas a cero.

Adelante y atrás para siempre

Si no le pasamos ningún parámetro, se mueve al 50% de velocidad

```
import cyberpi as cpi
cpi.mbot2.forward()
```

Si le paso un número como parámetro se mueve a esa velocidad:

```
import cyberpi as cpi
cpi.mbot2.forward(20)
```

Parar motores

```
import time
import cyberpi as cpi
cpi.mbot2.forward(20)
time.sleep(2)
cpi.mbot2.EM_stop(port = "all")
```

Con un segundo argumento le damos el tiempo en segundos:

```
import cyberpi as cpi
cpi.mbot2.forward(50,1)
cpi.mbot2.backward(50,1)
cpi.mbot2.forward(50,1)
cpi.mbot2.EM_stop(port = "all")
```

Adelante o atrás una cierta distancia

```
import cyberpi as cpi
cpi.mbot2.straight(40, speed = 50)
cpi.mbot2.straight(-40, speed = 50)
```

Girar sobre sí mismo

Para ello, las ruedas necesitan girar en diferente sentido:

```
import cyberpi as cpi
cpi.mbot2.turn_left(speed = 50, run_time = 1)
cpi.mbot2.turn_right(speed = 50, run_time = 1)
```

Girar en un ángulo concreto

```
import cyberpi as cpi
cpi.mbot2.turn(90, speed = 50)
```

Giro gradual

Si queremos hacer un giro mientras nos movemos, necesitamos girar la ruedas a diferente velocidad. controlar los motores del robot mBot2 (gestionado por la CyberPi) asignando un porcentaje de potencia específico a cada rueda.

La función `drive_power(izquierdo, derecho)` utiliza valores que van de -100 a 100: 60 (Motor Izquierdo): Gira hacia adelante al 60% de su potencia total. -40 (Motor Derecho): Gira hacia atrás al 40% de su potencia total (el signo negativo invierte el sentido). Resultado del Movimiento Al tener las ruedas girando en direcciones opuestas y con distintas fuerzas, el robot realizará un giro brusco hacia la derecha. No es un giro sobre su propio eje perfecto (sería 50, -50), sino un giro con un ligero desplazamiento. Ejemplos Comunes en Código Aquí tienes cómo se vería en un script de Python para mBlock: `Pythonimport cyberpi as cpi`

Ejemplo:

```
import cyberpi as cpi
cpi.mbot2.drive_power(60, -40) #left +, right -
time.sleep(2)
cpi.mbot2.EM_stop(port = "all")
```

Moverse hacia adelante

```
cpi.mbot2.drive_power(50, 50)
```

Realizar el giro que mencionaste

```
cpi.mbot2.drive_power(60, -40)
```

Detener los motores

`cpibot2.drive_power(0, 0)` Diferencias ClaveEs importante no confundir `drive_power` con `drive_speed`:FunciónUnidadPrecisióndrive_powerPorcentaje (-100 a 100)Varía según el nivel de batería.drive_speedcm/s (centímetros por segundo)Constante gracias a los encoders de los motores.

Plantillas de Código

Existen dos plantillas de código básicas que utilizamos al trabajar con motores. En ambos casos, usamos el botón A para activar el mBot2 e iniciar las acciones.

Separar el código en secciones hace que sea mucho más fácil de entender y de realizar cambios en él. Más adelante, añadiremos más secciones según las vayamos necesitando.

Programación de mBot2 / CyberPi con Python

1. Acciones Únicas (Single Actions)

Utiliza este bloque cuando quieras que las acciones del mBot2 ocurran solo una vez al iniciar el programa.

```
#IMPORTACIONES-----
import cyberpi as cpi
import time

#ESPERAR PARA EMPEZAR-----
cpi.console.println('Pulsa A')
while not cpi.controller.is_press('a'):
    cpi.led.on(255,0,0) # Rojo
    cpi.led.on(0,255,0) # Verde

#ACCIONES DEL ROBOT-----
cpi.mbot2.forward(speed = 50, run_time = 2) # Avanzar (velocidad 50, durante 2 seg)
cpi.mbot2.backward(speed = 50, run_time = 2) # Retroceder (velocidad 50, durante 2 seg)
cpi.led.off()
```

Si tenemos acciones que se repiten un número específico de veces, podemos usar un bucle for. Por ejemplo, para moverse en cuadrado:

```
#IMPORTACIONES-----
import cyberpi as cpi
import time

#ESPERAR PARA EMPEZAR-----
cpi.console.println('Pulsa A')
while not cpi.controller.is_press('a'):
    cpi.led.on(255,0,0)
    cpi.led.on(0,255,0)
```

```
#ACCIONES DEL ROBOT-----
# El robot hará un cuadrado (4 lados)
for i in range(4):
    cpi.mbot2.straight(40, speed = 50) # Avanza 40 cm
    cpi.mbot2.turn(90, speed = 50)     # Gira 90 grados

cpi.led.off()
```

2. Acciones Infinitas (Forever Actions)

Este código utiliza un bucle while True que repite las acciones indefinidamente, o hasta que presiones el botón "Home" junto a la conexión USB.

```
#IMPORTACIONES-----
import cyberpi as cpi
import time

#ESPERAR PARA EMPEZAR-----
cpi.console.println('Pulsa A')
while not cpi.controller.is_press('a'):
    cpi.led.on(255,0,0)
    cpi.led.on(0,255,0)

#BUCLE PRINCIPAL-----
while True:
    # El robot se moverá adelante y atrás continuamente
    cpi.mbot2.forward(speed = 50, run_time = 2) # Avanza (velocidad 50, 2 segundos)
    cpi.mbot2.backward(speed = 50, run_time = 2) # Retrocede (velocidad 50, 2 segundos)
```

Este tipo de código se utiliza principalmente en conjunto con el joystick, los botones o los sensores (ultrasónico y sigue-líneas), donde el mBot2 debe responder constantemente a los cambios en el entorno.

RETOS

Reto 1

- Coloca dos objetos pequeños en el suelo a una distancia mínima de 1 metro.
- Conduce alrededor de ellos varias veces formando un "8".
- Cuando gires, utiliza los LED para indicar tus giros.

Reto 2

- Coloca un objeto grande en el suelo y gira alrededor de él 3 veces en un círculo grande y suave (Utiliza la función cpi.mbot2.drive_power()).